AMENDMENTS TO THE SPECIFICATION

Please replace paragraph number [0048] with the following amended paragraph number [0048]:

[0048] FIG. 1 is a block diagram showing one embodiment of a system 10 for automated collection of document information which can be, for example, commercial mortgage loan information. A plurality of documents 1 are entered into a database 13 through an input device 3 such as, for example, an optical scanner or a graphical user interface residing on a PC. However, any form of data entry device capable of communicating with a server 5 can be used, as is known in the art. The database 13 resides on the server 5, where the database contains records ~~11~~ $11_1 - 11_n$ that store the plurality of documents 1. ~~The~~ Each of the individual records ~~11~~ $11_1 - 11_n$ are associated with individual assets. The database 13 may be a commercial relational database such as Microsoft® Access and can contain preprogrammed modules 9. The database 13 is capable of sorting, indexing and generating reports through output device 7 which may be, for example, a CD writer, a printer, a display on a PC, and the like. The database 13 includes a high-capacity storage medium configured to store individual asset records and related loan information. In one aspect, the database 13 is configured as a set of high speed, high-capacity hard drives, such as organized into a Redundant Array of Inexpensive Disks (RAID) volume, for example. However, any form of volatile storage, non-volatile storage, removable storage, fixed storage, random access storage, sequential access storage, permanent storage, erasable storage, and the like would be equally suitable.

Please replace paragraph number [0053] with the following amended paragraph number [0053]:

[0053] For each asset, the database 13 receives asset document information 15 pertaining to that asset, and the database module 9 stores the information 15 in record form in the database 13. The database module 9 organizes the asset document information 15 in the database 13. The database module 9 further provides the facilities for efficiently storing and accessing the asset information 15 that is collected on a flow basis. The asset document information is stored in a record ~~11~~ $11_1 - 11_n$ (as shown in FIG. 1) and organized into fields (an example of which is shown in FIG. 6). The database server 5 performs the functionality of the database module 9. Any type of database organization can be utilized, for example, a flat file system, hierarchical database, a relational database, or distributed database, such as provided by database vendors Microsoft® or Oracle®. In alternative embodiments of the present methods and systems, conventional spreadsheet software and Lotus Notes® applications can be employed.

Please replace paragraph number [0055] with the following amended paragraph number [0055]:

[0055] The aggregation module 21 collects data contained in a plurality of database records ~~11~~ $11_1 - 11_n$. A user of the system, for example, may desire information contained in one field of multiple database records. The aggregation module 21, working in conjunction with the database module 9, compiles the information contained in the various records into one datafile. The functionality of the aggregation module 21 is performed by the server 5.

Please replace paragraph number [0057] with the following amended paragraph number [0057]:

[0057] FIG. 3 is a block diagram showing the analysis module 23 of the server 5 of FIG. 2. The analysis module 23 contains three functional submodules: the database interface module 9E and the database query module 9B, and the comparison module 33. The database interface module 9E and query module 9B may be preprogrammed in a commercial database such as Microsoft® Access where a user can tailor them to specific needs, as is known in the art. The purpose of the comparison module 33 is to compare, for example, commercial mortgage loan document information 27, as shown, to a standard preprogrammed into the comparison module 33. This standard, as explained above, may be a banking industry standard or a legal standard. The comparison module 33 further compares the information 27 to information contained in the record ~~11~~ $11_1 - 11_n$ corresponding to that particular asset. The analysis module 23 can operate either in batch mode wherein an alert 37 is generated for information 27 from a group of asset documents or in a dynamic mode wherein the alert 37 is generated in real time as the information 27 is entered into the database 13.

Please replace paragraph number [0058] with the following amended paragraph number [0058]:

[0058] The analysis module 23 receives information 27 from, for example, commercial mortgage loan asset documents. This information is entered into the database 13 via a database interface module 9E. The interface module 9E calls the comparison module 33 based on a user selection of a particular asset. The comparison module 33, in turn, calls a database query via the database query module 9B that locates the appropriate record ~~11~~ 11_1 associated with the asset. The comparison module 33 compares the information 27 to a preprogrammed standard, a banking standard or legal standard, for example. If the information 27 does not meet that standard, the alert 37 is generated that is presented to a user via the interface module 9E. The standard may be programmed in the form of a user-selected template for data entry into a particular asset record, or a software object called by the comparison module 33 with preprogrammed logic reflecting the appropriate standard. If the information 27 is already contained in the database in another field of the record ~~11~~ 11_1, the comparison module 33 pulls this preexisting information via the query module 9B and compares it to information 27. For example, with regard to commercial mortgage loans, building square footage contained in the mortgage document can be compared against the square footage in the building inspection. The comparison module compares the square footage information contained in the record ~~11~~ 11_1 and, in the event of a mismatch, an alert 37 can be generated and displayed to the user.

Please replace paragraph number [0059] with the following amended paragraph number [0059]:

[0059] FIG. 4 is a block diagram showing the extraction module 19 of the server system in FIG. 2. The extraction module 19 contains three submodules: the database interface module 9E, the information extraction and format module 45 and the database query module 9B. The database interface module 9E and query module 9B function in the manner stated above with respect to the analysis module 23. An information request 49 is sent to the extraction and format module 45 via the interface module 9E. The extraction and format module 45 calls the database query module 9B to find the appropriate record ~~11~~ 11_2 in the database (not shown). The extraction and format module 45 extracts the requested information 49 from the record into a temporary datafile and then formats the extracted data in a manner requested by the user. The interface module 9E may contain preprogrammed graphical user interface screens programmed to display the extracted and formatted information. The interface module 9E may further contain options to print the extracted information, for example, or send the information to a CD or electronic mail.

Please replace paragraph number [0060] with the following amended paragraph number [0060]:

[0060] FIG. 5 is a block diagram showing the aggregation module 21 of the server system in FIG. 2. The aggregation module 21 collects data contained in a plurality of database records ~~11~~ $11_1 - 11_n$. A user of the system, for example, may desire common information contained, for example, in one field of multiple database records. The user generates a request for information 57 that is sent to the information compilation module 59 via the database interface module 9E. The information compilation module 59, in turn, calls the query module 9B. The compilation module 59 and the query module 9B determine what records ~~11~~ $11_1 - 11_n$ must be queried and collected to satisfy the request 57. The query module 9B indexes the database based upon the request 57. Upon indexing of the records containing the requested common information, the compilation module 59 extracts the data contained in the fields of the records ~~11~~ $11_1 - 11_n$ and writes it to a temporary datafile. An example of this is a request by a user for all property balance sheets contained in a plurality of database records. The compilation module 59, in conjunction with the query module 9B, indexes the database on the field of property balance sheets. The datafile is then displayed via the interface module 9E in the form of aggregated common information 51 as requested.

Please replace paragraph number [0061] with the following amended paragraph number [0061]:

[0061] FIG. 6 is a record view illustrating the fields comprising an example record ~~11~~ 11_n shown in the database 13 of the system of FIG. 1. Each record ~~11~~ 11_n is illustrative of some of the information typically required to securitize an asset or loan into a trust or SPV. For example, for a commercial mortgage loan, information associated with the following documents can be recorded in the database record ~~11~~ 11_n: the asset name 61, appraisal 63, promissory note 67, escrow agreement 71, rent roll 75, and operating statement 79. Pertinent information is taken from documents 63, 67, 71, 75, and 79 and entered into their corresponding representative data fields 65, 69, 73, 77, and 81, respectively. The list is illustrative, but not exhaustive, of the information typically required for a commercial mortgage loan and can be modified by one skilled in the art to accommodate other fields for any financial instruments suitable for securitization. From these fields, many forms of investor information can be generated as is known in the art. The fields shown in FIG. 6 can be common to two or more records contained in the database 13 and can serve as an index for the database query module 9B. The records are typically indexed based on field 61 and stored in that manner. The information compilation module 59 can pull information contained in a single field or multiple fields.

Please replace paragraph number [0062] with the following amended paragraph number [0062]:

[0062] Common information can be stored in more than one field of the record ~~11~~ 11_n. For example, the appraisal value contained in appraisal field 63 must match the appraisal value entered as part of the promissory note information 67 entered in representative data fields 69. The comparison module 33 described above compares the information stored in these fields upon entry of data into either field and generates the alert 37 if the appraisal value does not match.

Please replace paragraph number [0063] with the following amended paragraph number [0063]:

[0063] FIG. 7 is a flow diagram showing one embodiment of a method for inputting data into either an existing record or creating a record and then inputting data on a flow basis. The input data is compared to preexisting data and to a standard. In one aspect, the method can be implemented as a conventional computer program for execution by the server system 10 shown in FIG. 1. As a preparatory step, the user determines in step 95 if this is a preexisting asset matching a record ~~11~~ $11_1 - 11_n$ in the database 13 (as shown in FIG. 1). If the record exists, the database executes a query and retrieves the record in step 99. The user then inputs the asset data in step 101 into the record. The entered data is then compared to preexisting data in step 105 to determine if there is a data mismatch in step 107. In the event of a mismatch, the user is alerted in step 109. If the record does not exist, the user creates the record in step 97 corresponding to the asset and then inputs the data in step 101 into the database. Upon successful data entry, the entered data is compared to a standard in step 111 to ensure compliance with the standard. Upon comparison to the standard in step 113, the program generates, for example, an alert, a report or other like notification if the data does not meet the standard in step 115. The asset documents are entered into the database as soon as they become available to a user, or in a reasonable time thereafter, and are compared to a standard and preexisting data as the asset information is entered into the database. Thus, it can be seen that the information is flowed into the database or entered on a flow basis.

Please replace paragraph number [0065] with the following amended paragraph number [0065]:

[0065] FIG. 9 is a flow diagram showing one embodiment of a routine for extracting electronic facsimiles of asset documents stored in a record. The purpose of this routine is to extract copies of original documents stored in a record ~~11~~ $11_1 - 11_n$ associated with an asset in a database 13. The user, through the database interface, requests copies of original asset documents in step 133. The user selects the applicable asset in step 135. The appropriate record associated with the asset is queried and retrieved from the database in step 137. The required documents are extracted from the retrieved records in step 139 and displayed as a list to the user in step 140. The user selects the means for extracting the electronic facsimiles of the documents in step 141, where the means include, for example: displaying on a monitor in step 143, hard copy printing in step 147 or writing the information to a CD, electronic mail or website in step 145, for example.

Please replace paragraph number [0067] with the following amended paragraph number [0067]:

[0067] FIG. 11 is a block diagram showing one illustrative embodiment of the hardware components of the system shown in FIG. 1. A remote PC 179 can be equipped with a printer 181A, B, C a scanner 177A, B, C and a CD optical drive 175A, B, C. The components comprise a workstation 169 capable of communicating with the Internet 171. Each workstation is preferably equipped with 64MB RAM, 10GB hard drive capacity and related equipment. In addition, the workstations 169A, B, C are also Intel Pentium®-based personal computer workstations available from, for example, Dell or Compaq. The workstations 169A, B, C ~~and 167~~ each further comprise a programmed digital computing device consisting of a central processing unit (CPU), random access memory (RAM), non-volatile secondary storage, such as a hard drive or CD-ROM drive, network interfaces, and peripheral devices, including user interface means, such as keyboard and display. Program code, including software programs and data are loaded into the RAM for execution and processing by the CPU and results are generated for display, output, transmittal, or storage. The server 5 is a general purpose, programmed digital computing device consisting of a central processing unit (CPU), random access memory (RAM), non-volatile secondary storage, such as a hard drive or CD-ROM drive, network interface, and peripheral devices, including user interface means, such as keyboard and display. Program code, including software programs and data are loaded into the RAM for execution and processing by the CPU and results are generated for display, output, transmittal, or storage. In the described embodiment, the individual server is an Intel-Pentium® based server system such as is available from Dell or Compaq. The system is preferably equipped with at least 128MB RAM, and at least 100GB hard drive capacity, data backup facilities, and related hardware for interconnection to an intranet and the Internet. Other types of server and workstation systems including minicomputers, mainframe computers, supercomputers, parallel computers, digital data processors and the like are equally suitable, as is known in the art.

Please replace paragraph number [0069] with the following amended paragraph number [0069]:

[0069] In the present embodiment, asset data is input via workstations 169 and transmitted via the Internet 170 through the connection 172 and to the intranet server 173. Information is also input through workstations ~~167~~ 169C connected directly to the intranet server 173. The information can be routed 174 to the database server 5 for storage in the database 13. To ensure reliable data exchange, the intranet server can implement a TCP/IP protocol stack, although other forms of network protocol stacks, such as HTTP, are suitable.

Please replace paragraph number [0070] with the following amended paragraph number [0070]:

[0070] Referring now to Figures 12A and 12B, FIG. 12A is an example of a graphical user interface of one embodiment of the system shown in FIG. 1. As shown, the screen 182 contains a directory tree 185 that corresponds, in part, to the fields of the database record ~~11~~ 11_n (as shown in FIG. 6) and is an example of one embodiment of the database interface module 9E shown in Figures 3, 4 and 5. A dialogue box 217 shows an example of a further embodiment of the interface module 9E where information is entered into a field of the record ~~11~~ 11_n. The dialogue box 217, for example, demonstrates the user selecting the appropriate asset 183 associated with the record and by this action the program sets up an automatic scheduler 218, that can be used, for example, to alert the user to periodically obtain new financial statements from the borrower as required by the loan agreement. The system can further create individual tracking records for each asset document entered.

Please replace paragraph number [0073] with the following amended paragraph number [0073]:

[0073] FIG. 16 is a graphical user interface for viewing an electronic facsimile of an asset document scanned into the database. The dialogue box 223, as shown in FIG. 14, allows a user to view an electronic facsimile of an asset document 227 after the document is scanned into the system. It is appreciated from the screen view 182 and further from the directory tree 185 that any image stored for a given field of record ~~11~~ 11_1 - 11_n is capable of viewing via dialogue box 223. A user selects an asset 183 causing a directory tree 185 for that asset to display. The user then causes the dialogue box 223 to display and selects image to view the asset document associated with that field in the record ~~11~~ 11_1 - 11_n associated with asset 183.

Please replace paragraph number [0075] with the following amended paragraph number [0075]:

[0075] FIG. 18 is an example of a graphical user interface for selecting a loan portfolio or set of loans to produce a report or datafile. The user causes dialogue box 237 to display. In one embodiment, the dialogue box 237 accepts user input 239 and causes a module to execute that aggregates data in accordance with user input 239. In one embodiment, for example, the user requests data derived from source asset documents contained in a series of assets (records ~~11~~ 11_1 - 11_n), for example, all loan terms and property values for a collection of assets. The program, utilizing the compilation module 59 as shown in FIG. 5, generates a datafile that presents loan terms as stated in each promissory note and property values stated in each appraisal.

Please replace paragraph number [0076] with the following amended paragraph number [0076]:

[0076] FIG. 19 is an example of a graphical user interface for selecting a report from a user selection shown in FIG. 18. The datafile produced in FIG. 18 is presented to the user in a dialogue box 243 and contains a series of preprogrammed reports 244. The user has the option of viewing a report from the list 244, sending the report via e-mail 245 or printing the report. The directory tree 241 comprises a list of assets, each asset corresponding to a record ~~11~~ $\underline{11_1 - 11_n}$ in the database 13.